IJCTA, 9(5), 2016, pp. 2651-2675

New Methodology of Online Sliding Surface Slope Tuning PID like Fuzzy Sliding Mode Controller for Robust Control of Robot Manipulators

S. Yadegar[1], A. CheSoh[2], S. A. Ahmad[3] and A. J. Ishak[4]

ABSTRACT

Designing a robust controller for uncertain multi input-multi output (MIMO) nonlinear dynamical system (e.g., robot manipulator) can be a challenging work in this research. Robot manipulators are set of links which connected by joints, they are multiple-input and multiple-output (MIMO), nonlinear, time variant, uncertain dynamic systems and are developed either to replace human work in many fields such as in industrial or in the manufacturing. According to the dynamic formulation of robot manipulators, they are uncertain and have strong coupling effects between joints. To solve this challenge, sliding mode controller is selected since it is robust, stable and it works very well in certain and fairly uncertain condition. Although this controller works incredibly efficient, but still, it has two important challenges, namely the high frequency chattering and working in uncertain situation. To reduce the chattering with respect to stability and robustness; linear controller is added to discontinuous (switching) part of sliding mode controller. In this methodology linear controller is used in parallel with discontinuous part to reduce the role of sliding surface slope and switching (sign) function. To modify chattering free sliding mode controller in uncertain situation PID like fuzzy logic theory is recommended in estimating the robot manipulator's nonlinear dynamic formulation and on-line tuning sliding surface slope.As a result, this controller improves the stability and robustness, reduces the chattering as well and reduces the level of energy due to the torque performance as well.

Keywords: PID like fuzzy controller, robust sliding mode controller, online tuning, chattering phenomenon, robot manipulator

1. INTRODUCTION

PUMA robot manipulator is a serial link six degrees of freedom manipulator which the dynamics formulation is highly nonlinear, time variant, MIMO, uncertain and have strong coupling effects between joints [1-6]. Designing a linear behavioural controller in order to reduce or cancel decoupling as well as to achieve stability, robustness and reliability are the ultimate objectives of this research. To achieve these goals, in the first part linear controller is investigated but it has two limitations; the first limitation is reducing the output velocity and acceleration while the second one is the need to design a system based on high gear ratio.

Therefore linear type controller, such as PD or PID cannot accomplish a good performance. To achieve acceptable velocity and acceleration, linearization and decoupling without using many gears, nonlinear

[1] Research Scholar, Depart. of Electrical & Electronic Engineering, Universiti Putra Malaysia, Serdang, Malaysia, *Email: sanaz.yadegar@yahoo.com*

[2] Research Supervisor, Research Scholar, Depart. of Electrical & Electronic Engineering, Universiti Putra Malaysia, *Email: azuracs@upm.edu.my*

[3] Research Co-Supervisor, Research Scholar, Depart. of Electrical & Electronic Engineering, Universiti Putra Malaysia, *Email: sanom@upm.edu.my*

[4] Research Co-Supervisor, Research Scholar, Depart. of Electrical & Electronic Engineering, Universiti Putra Malaysia, *Email: asnorji@upm.edu.my*

control methodology is investigated [7-11]. One of the nonlinear control methodology is the computed torque controller.This controller works based on acceleration measurements but this method is very expensive, and that is the first challenge. While the second challenge is related to the accuracy of modelling robot manipulator’s dynamic. It is very difficult to include all the complexities in the system dynamic to obtain the accurate model. To eliminate the actual acceleration measurement and also the accurate computation ‘burden of PUMA robot dynamic’s as well as to achieve stability, efficiency and robustness in controller, sliding mode controller technique is evaluated in the following part [1-2].

Conventional sliding mode controller (SMC) is a reliable nonlinear controller, model base, stable and robust. To design a controller with the presence of uncertainties and external disturbance, this controller is a strong candidate. Even though this type of controller is working in many applications but there are three main issues limiting the applications of conventional sliding mode controller which are (i) equivalent part related to dynamic equation of robot manipulator, (ii) chattering phenomenon and (iii) computation the uncertainties problem in chattering free equivalent estimation sliding mode controller [12-25].

The first challenge to design and apply the sliding mode controller for robot manipulator is the equivalent part related to highly nonlinear dynamic equation. This problem is not a simple challenge. In order to solve this, fuzzy logic is used as parallel controller with conventional sliding mode controller acting as the model-base fuzzy sliding mode controller. In model-base fuzzy sliding mode controller, fuzzy logic controller is used as an estimator to eliminate the dynamic uncertainties. To design fuzzy logic controller, PID like fuzzy logic controller is evaluated. PID like fuzzy logic controller has three inputs, Proportional (P), Derivative (D), and Integral (I), if each input has N linguistic variables to define the dynamic behavior, it has $N \times N \times N$ linguistic variables. There will be too many works to be carried out to define and write N^3 rules. Hence, the speed of controller will be too low, and the design and implementation of the embedded controller based on micro-based technology can be very difficult. To solve the number of rule base in fuzzy model-base sliding mode controller parallel strategy is evaluated based on a parallel structure of a PD-like fuzzy controller and PI-like fuzzy controller. Hence, the challenge of designing PI and PD fuzzy rule tables are supposed to be solved. However, designing two types rule tables are very difficult. The most important reason to select PD type rule table is because of designing PI rule table needs to the very wide universe of discourse. To solve this challenge PID-like fuzzy controller is replaced by PD-like fuzzy controller, where the integral term is in the output. This method is the reason why only PD type rule table need to be designed for PD-like fuzzy controller and PI-like fuzzy controller.

The second challenge in designing a robust sliding mode based controller is the chattering phenomenon and this problem can be caused by heating and oscillation in mechanical part of the system. The main objective in this study is basically to reduce or eliminate this chattering phenomenon in order to maintain the robustness of the system. Switching function can cause chattering but it is one of the main parts in robust sliding mode controller design. The sliding surface slope (λ) is the second factor that contributes to chattering in sliding mode controller. Therefore, the main task in this section is to reduce or eliminate the chattering in conventional robust sliding mode controller based on design parallel linear control methodology and discontinuous part [26-31]. Based on Lyapunov theory, conventional sliding mode controller and linear control methodologies are robust. Therefore based on switching theory, Lyapunov stability has been proven in the proposed chattering-free sliding mode controller.

Uncertainties are very important challenge and cause extremely high estimation of the bounds [27-28]. To solve this problem, selecting the desired sliding surface and $sign$ function plays a vital role and if the dynamic of robot manipulator is derived to sliding surface then the linearization and decoupling through the use of feedback, not gears, can be realized. In this state, the derivative of sliding surface can help to decouple and linearize closed-loop PUMA robot dynamics that one expects in computed torque control. Linearization and decoupling by the above method can be obtained in spite of the quality of the dynamic model of robot manipulator, in contrast with the computed-torque control that requires accurate dynamic model of the system. It is well known fact that if the uncertainties are very good compensated there is no need to use discontinuous part which creates the chattering [29-33]. To compensate the uncertainties fuzzy logic theory is a good candidate, but design a fuzzy controller with perfect dynamic compensation in the presence of uncertainty is very difficult. Therefore, in this research uncertainties are estimated by discontinuous feedback control and the linear part controller is added to this part to eliminate the chattering. To increase the bound of uncertainty fuzzy gains and sliding surface slope coefficients will be tuned via online tuning method. The above discussion gives rational in selecting the proposed methodology in this research.

This research is organized as follows; the second part is focusing on the modeling of dynamic formulation based on Lagrange methodology. The third part is focusing on the method of reducing error, increasing the performance quality and increasing the robustness as well as stability. The simulation results and discussions are illustrated in the fourth part on the subject of trajectory following and disturbance rejection. The last part focuses on the conclusions and comparisons between this method and few other methods.

2. DYNAMIC MODELING OF ROBOT MANIPULATOR

A dynamic function is the study of correlation between motion and forces. Dynamic modeling is used to illustrate the behavior of robot manipulators (e.g., nonlinear dynamic behavior), design of nonlinear conventional controller (e.g., conventional computed torque controller, conventional sliding mode controller and conventional backstepping controller) and for simulation. It is used to analyse the relationship between dynamic functions output (e.g., joint motion, velocity, and accelerations) and input source of dynamic functions (e.g., force/torque or current/voltage). Dynamic functions is also used to explain some dynamic parameter's effect (e.g., inertial matrix, Coriolis, Centrifugal, and some other parameters) of a system's behavior [3]. The equation of a multi degrees of freedom (DOF) robot manipulator is considered by the following equation[7-10]:

$$[A(q)]\ddot{q}+[N(q,\dot{q})]=[\tau] \tag{1}$$

where τ is actuator's torque with $n \times 1$ vector, A (q) is positive define inertia with $n \times n$ symmetric matrix based on the following formulation;

$$A(q)=\begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & \cdots & A_{1n} \\ A_{21} & \cdots & \cdots & \cdots & \cdots & A_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{n.1} & \cdots & \cdots & \cdots & \cdots & A_{n.n} \end{bmatrix} \tag{2}$$

$N(q,\dot{q})$ is the vector of nonlinearity term, while q is $n \times 1$ joints variables. If all joints are revolute, the joint variables are angle (θ) and if these joints are translated, the joint variables are translating position (d). According to (1) the nonlinearity term of robot manipulator is derived as three main parts; Coriolis $b(q)$**,** Centrifugal $C(q)$, and Gravity $G(q)$. Hence, the dynamic equation of the robot manipulator can be written as [8]:

$$[N(q,\dot{q})]=[V(q,\dot{q})]+[G(q)] \tag{3}$$

$$[V(q,\dot{q})]=[b(q)],[\dot{q}\dot{q}]+[C(q)][\dot{q}]^2 \tag{4}$$

$$\tau = A(q)\ddot{q} + b(q)][\dot{q}\dot{q}] + C(q)][\dot{q}]^2 + G(q) \tag{5}$$

where, $b(q)$ is a Coriolis torque matrix and is $n \times \frac{n \times (n-1)}{2}$ matrix, $C(q)$ is Centrifugal torque matrix and is $n \times n$ matrix, Gravity is the force of gravity and is $n \times 1$ matrix, $[\dot{q}\dot{q}]$ is vector of joint velocity that it can give by: $[\dot{q}_1.\dot{q}_2, \dot{q}_1.\dot{q}_3, ..., \dot{q}_1.\dot{q}_n, \dot{q}_2.\dot{q}_3, ...]^T$, and $[\dot{q}]^2$ is vector, that it can given by:

$\left[\dot{q}_1^2, \dot{q}_2^2, \dot{q}_3^2, ...\right]^T$. According to the basic information from university all functions are derived as the following form;

$$Outputs = function\ (inputs) \tag{6}$$

In the dynamic formulation of robot manipulator the inputs are torques matrix and the outputs are actual joint variables, consequently (7) is derived as (6);

$$q = function\ (\tau) \tag{7}$$

$$\ddot{q} = A^{-1}(q).\{\tau - N(q, \dot{q})\} \tag{8}$$

$$q = \iint A^{-1}(q).\left\{\tau - N(q, \dot{q})\right\} \tag{9}$$

The Coriolis matrix (b) is a $n \times \frac{n(n-1)}{2}$ matrix which calculated as follows;

$$b(q) = \begin{bmatrix} b_{112} & b_{113} & \cdots & b_{11n} & b_{123} & \cdots & b_{12n} & \cdots & \cdots & b_{1.n-1.n} \\ b_{212} & \cdots & \cdots & b_{21n} & b_{223} & \cdots & \cdots & \cdots & \cdots & b_{1.n-1.n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ b_{n.1.2} & \cdots & \cdots & b_{n.1.n} & \cdots & \cdots & \cdots & \cdots & \cdots & b_{n.n-1.n} \end{bmatrix} \tag{10}$$

The Centrifugal matrix (C) is $n \times n$ a matrix;

$$C(q) = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{bmatrix} \tag{11}$$

The Gravity vector (G) is a $n \times 1$ vector;

$$G(q) = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \tag{12}$$

According to [8-11], the dynamic formulations of six Degrees of Freedom serial links PUMA robot manipulator are computed by;

$$A(\ddot{\theta})\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\\ \ddot{\theta}_3\\ \ddot{\theta}_4\\ \ddot{\theta}_5\\ \ddot{\theta}_6\end{bmatrix}+B(\theta)\begin{bmatrix}\dot{\theta}_1\dot{\theta}_2\\ \dot{\theta}_1\dot{\theta}_3\\ \dot{\theta}_1\dot{\theta}_4\\ \dot{\theta}_1\dot{\theta}_5\\ \dot{\theta}_1\dot{\theta}_6\\ \dot{\theta}_2\dot{\theta}_3\\ \dot{\theta}_2\dot{\theta}_4\\ \dot{\theta}_2\dot{\theta}_5\\ \dot{\theta}_2\dot{\theta}_6\\ \dot{\theta}_3\dot{\theta}_4\\ \dot{\theta}_3\dot{\theta}_5\\ \dot{\theta}_3\dot{\theta}_6\\ \dot{\theta}_4\dot{\theta}_5\\ \dot{\theta}_4\dot{\theta}_6\\ \dot{\theta}_5\dot{\theta}_6\end{bmatrix}+C(\theta)\begin{bmatrix}\dot{\theta}_1^2\\ \dot{\theta}_2^2\\ \dot{\theta}_3^2\\ \dot{\theta}_4^2\\ \dot{\theta}_5^2\\ \dot{\theta}_6^2\end{bmatrix}+G(\theta)=\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\\ \tau_5\\ \tau_6\end{bmatrix} \tag{13}$$

where

$$A(q)=\begin{bmatrix}A_{11} & A_{12} & A_{13} & 0 & 0 & 0\\ A_{21} & A_{22} & A_{23} & 0 & 0 & 0\\ A_{31} & A_{32} & A_{33} & 0 & A_{35} & 0\\ 0 & 0 & 0 & A_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & A_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & A_{66}\end{bmatrix} \tag{14}$$

Based on [8] the Coriolis () matrix elements are;

$$b(q)=\begin{bmatrix}b_{112} & b_{113} & 0 & b_{115} & 0 & b_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & b_{214} & 0 & 0 & b_{223} & 0 & b_{225} & 0 & 0 & b_{235} & 0 & 0 & 0 & 0\\ 0 & 0 & b_{314} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ b_{412} & b_{413} & 0 & b_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & b_{514} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{15}$$

According to [8] Centrifugal () matrix elements are;

$$C(q) = \begin{bmatrix} 0 & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & 0 & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ C_{51} & C_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

Gravity (G) Matrix elements are [8];

$$[G(q)]_{6\times1} = \begin{bmatrix} 0 \\ G_2 \\ G_3 \\ 0 \\ G_5 \\ 0 \end{bmatrix} \tag{17}$$

3. METHODOLOGY

Sliding Mode Controller (SMC) is a robust and nonlinear conventional controller in a partially uncertain parameters of a dynamic system. Applications such as in aerospace, robotics, process control and many more are using this conventional nonlinear controller. Using this controller can resolve the challenge of stability and robustness in control theory [18-22, 34-41]. The main idea to design sliding mode control is based on the following formulation;

$$\tau(q,t) = \begin{cases} \tau_i^+(q,t) & if \quad S_i > 0 \\ \tau_i^-(q,t) & if \quad S_i < 0 \end{cases} \tag{18}$$

where S_i is sliding surface (switching surface), $i = 1, 2, \ldots \ldots, n$ for n-DOF robot manipulator, $\tau_i(q, t)$ is the i^{th} torque of joint. The dynamic formulation of nonlinear single input system can be defined as [7, 42-48]:

$$x^{(n)} = f(\vec{x}) + b(\vec{x})u \tag{19}$$

In Eq. (19) u is the vector of control input, $x^{(n)}$ is the n^{th} derivation of x, $x = [x, \dot{x}, \ddot{x}, \ldots, x^{(n-1)}]^T$ is the state vector, $f(x)$ is unknown or uncertainty, and $b(x)$ is known *switching (SIGN)* function. Designing a sliding mode controller is mainly to achieve high speed train and high tracking accuracy to the desired joint variables; $x_d = [x_d, \dot{x}_d, \ddot{x}_d, \ldots, x_d{}^{(n-1)}]^T$, according to actual and desired joint variables, the trucking error vector is defined by [7]:

$$\tilde{x} = x_d - x_a = [\tilde{x}, \ldots, \tilde{x}^{(n-1)}]^T \tag{20}$$

According to the theory of sliding mode controller, the main important part to design this controller is the sliding surface, where the time-varying sliding surface $s(x, t)$ in the state space R^n is given by the following formulation [7]:

$$s(x,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} = 0 \tag{21}$$

Based on Eq. (21) λ is the sliding surface slope coefficient and it is a positive constant. The sliding surface can be defined as Proportional-Derivative (PD), Proportional-Integral (PI) and the Proportional-Integral-Derivative (PID). The following formulations represented the three groups are [7]:

$$S_{PD} = \lambda e + \dot{e} \tag{22}$$

$$s(x,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \left(\int_0^t \tilde{x}\, dt\right) = 0 \tag{23}$$

$$S_{PI} = \lambda e + \left(\frac{\lambda}{2}\right)^2 \int e \tag{24}$$

$$S_{PID} = \lambda e + \dot{e} + \left(\frac{\lambda}{2}\right)^2 \int e \tag{25}$$

Integral part of sliding surface is used to decrease the steady state error in sliding mode controller. In order to gain stability and minimum error in sliding mode controller, the sliding surface slope $s(x, t)$ is kept near to the zero. One of the common strategies is by finding input U outside of $s(x, t)$ [49-56].

$$\frac{1}{2}\frac{d}{dt} s^2(x,t) \le -\zeta \left|s(x,t)\right| \tag{26}$$

In Eq. (26) ζ is a positive constant.

$$\text{If} \quad S(0) > 0 \rightarrow \dot{S}(t) \le -\zeta \tag{27}$$

In Eq. (27) derivative term of (s) is eliminated by limited integral from $t = 0$ to $t = t_{reach}$

$$\int_{t=0}^{t=t_{reach}} \dot{S}(t) \le -\int_{t=0}^{t=t_{reach}} \eta \rightarrow S(t_{reach}) - S(0) \le -\zeta(t_{reach} - 0) \tag{28}$$

In Eq. (28) t_{reach} is the time that trajectories reach the sliding surface. If St_{reach} = 0 the formulation of t_{reach} calculatedby;

$$0 - S(0) \le -\eta(t_{reach}) \rightarrow t_{reach} \le \frac{S(0)}{\zeta} \tag{29}$$

In (29) if $S(0) < 0$

$$0 - S(0) \le -\eta(t_{reach}) \rightarrow S(0) \le -\zeta(t_{reach}) \rightarrow t_{reach} \le \frac{|S(0)|}{\eta} \tag{30}$$

According to Eq. (29) the formulation of Eq. (30) ensures that time to reach the sliding surface is smaller than $\frac{|S(0)|}{\zeta}$ since the trajectories are outside of $S(t)$.

$$f \; S_{t_{reach}} = S(0) \rightarrow error \;\; (x - x_d) = 0 \tag{31}$$

According to the above discussion, formulation of sliding surface (S) can be defined as

$$s(x,t) = \left(\frac{d}{dt} + \lambda\right)\tilde{x} = (\dot{x} - \dot{x}_d) + \lambda(x - x_d) \tag{32}$$

The change of sliding surface $(\dot{S})$ is;

$$\dot{S} = (\ddot{x} - \ddot{x}_d) + \lambda(\dot{x} = \dot{x}_d) \tag{33}$$

According to the formulation of the second order system, one easy solution to get the sliding condition when the system dynamics have uncertain parameters or external disturbance is by manipulating the switching control law, where:

$$U_{dis} = K(\vec{x}, t).\mathrm{sgn}(s) \tag{34}$$

In Eq. (34) the switching function sgn(s) is defined as [3, 7]

$$\mathrm{sgn}(s) = \begin{cases} 1 & s > 0 \\ -1 & s < 0 \\ 0 & s = 0 \end{cases} \tag{35}$$

In Eq. (34) the $K(\vec{x}, t)$ is the positive constant and based on Eqs. (22), (24) and (25) the sliding surface can be PD, PI and PID. According to above formulation, the formulation of sliding mode controller for robot manipulator is [3, 7];

$$\tau = \tau_{eq} + \tau_{dis} \tag{36}$$

In Eq.(36) τ_{eq} is equivalent term of sliding mode controller and this term is related to the nonlinear dynamic formulation of a robot manipulator. Conventional sliding mode controller is reliable controller based on the nonlinear dynamic formulation (equivalent part). The switching discontinuous part is introduced by τ_{dis} which are the important factors for the controller in terms of resistance and robustness. In serial links of six axes robot manipulator the equivalent part is written as follows;

$$\tau_{eq} = \left[A^{-1}(q) \times \left(N(q, \dot{q})\right) + \dot{S}\right] \times A(q) \tag{37}$$

In Eq. (37) the nonlinear term of $N(q, \dot{q})$ is;

$$\left[N(q, \dot{q})\right] = \left[V(q, \dot{q})\right] + \left[G(q)\right] \tag{38}$$

$$\left[V(q, \dot{q})\right] = \left[b(q)\right][\dot{q}\dot{q}] + \left[C(q)\right][\dot{q}]^2 \tag{39}$$

In PD sliding surface, based on Eq. (22) the change of sliding surface is calculated as;

$$S_{PD} = \lambda e + \dot{e} \rightarrow \dot{S}_{PD} = \lambda \dot{e} + \ddot{e} \tag{40}$$

While the discontinuous switching term (τ_{dis}) is computed as [3];

$$\tau_{dis} = K \cdot \mathrm{sgn}(S) \tag{41}$$

Based on Eq. (41) and Eq. (22);

$$\tau_{dis-PD} = K \cdot \mathrm{sgn}(\lambda e + \dot{e}) \tag{42}$$

According to Eq.(41) and Eq. (24);

$$\tau_{dis-PI} = K \cdot \mathrm{sgn}\left(\lambda e + \left(\frac{\lambda}{2}\right)^2 \int e\right) \tag{43}$$

By replacing (25) in (41) the discontinuous switching part will be;

$$\tau_{dis-PID} = K \cdot \mathrm{sgn}\left(\lambda e + \dot{e} + \left(\frac{\lambda}{2}\right)^2 \int e\right) \tag{44}$$

According toEqs. (37) and (41);

$$\tau = \tau_{eq} + K \cdot \mathrm{sgn}(S) = \left[A^{-1}(q) \times \left(N(q,\dot{q})\right) + \dot{S}\right] \times A(q) + K \cdot \mathrm{sgn}(S) \tag{45}$$

According to Eqs. (42) and (45) the formulation of PD-SMC is;

$$\tau_{PD-SMC} = K \cdot \mathrm{sgn}(\lambda e + \dot{e}) + \left[A^{-1}(q) \times \left(N(q,\dot{q})\right) + \dot{S}\right] \times A(q) \tag{46}$$

Figure 1 shows the PD sliding mode controller for serial links robot manipulator.

In sliding mode controller selecting the desired sliding surface and *sign* function have a significant effect to system performance and if the dynamic of robot manipulator is derived to sliding surface then the linearization and decoupling through the use of feedback, not gears, can be realized. In this state, the derivative of sliding surface can help to decoupled and linearized closed-loop PUMA robot dynamics that one expects in computed torque control. Linearization and decoupling by sliding mode controller can be obtained in spite of the poor quality of the robot manipulator dynamic model, in contrast to the computed-torque control that requires the exact dynamic model of a system. As a result, uncertainties are estimated by discontinuous feedback control but it can cause to chattering. To reduce the chattering in presence of switching functions; linear controller is added to discontinuous part of sliding mode controller. Linear controller is one type of stable controller and so is the conventional sliding mode controller. In the proposed

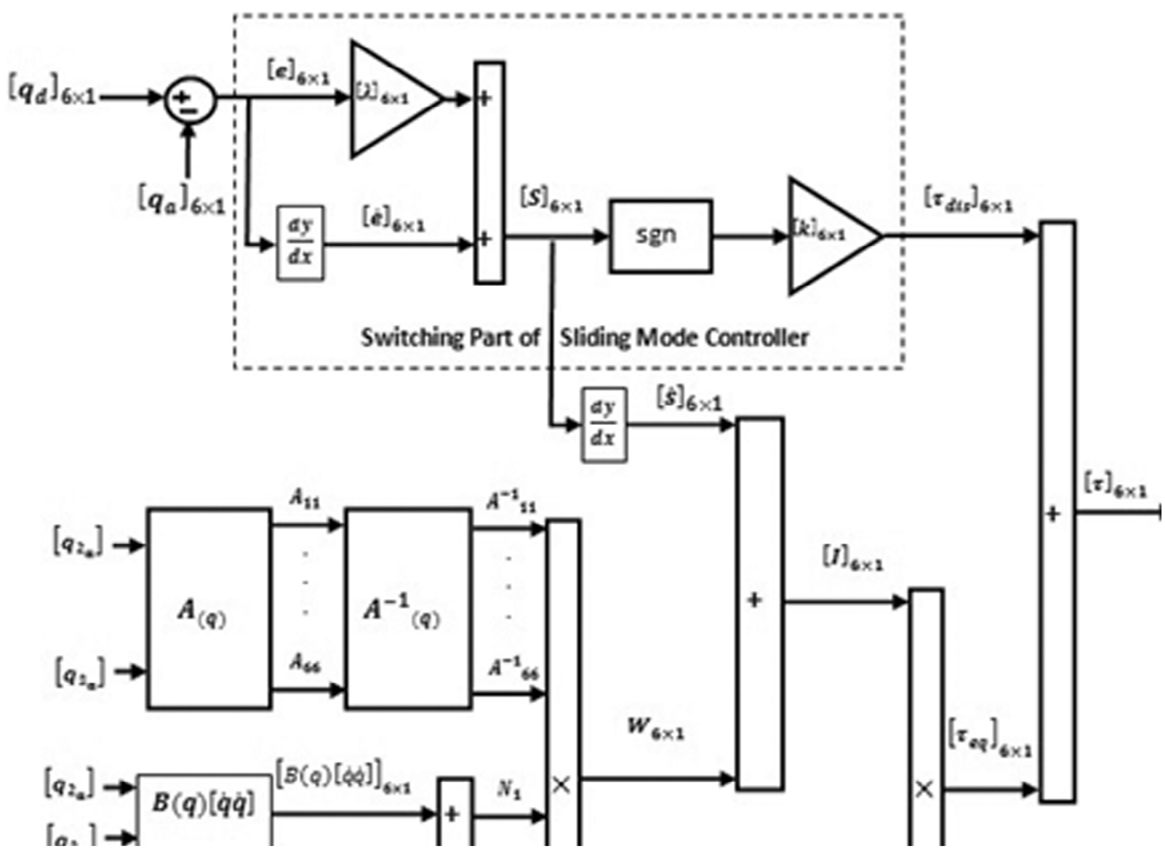

Figure 1: Block diagram of PD sliding mode controller for robot manipulator

methodology, PD, PI or PID linear controller is used in parallel with discontinuous part to reduce the role of sliding surface slope as main coefficient. The formulation of new chattering free sliding mode controller for robot manipulator is recommended as;

$$\tau = \tau_{eq} + \tau_{dis-new} \tag{47}$$

In Eq. (47) τ_{eq} is equivalent term of sliding mode controller and this term is related to the robot manipulator's nonlinear dynamic formulation. The new switching discontinuous part is introduced by $\tau_{dis-new}$ and this item is the important factor to resistance and robustness in this controller. In PD sliding surface, based on Eq. (5) the change of sliding surface is calculated as;

$$S_{PD} = \lambda e + \dot{e} \rightarrow \dot{S}_{PD} = \lambda \dot{e} + \ddot{e} \tag{48}$$

The discontinuous switching term (τ_{dis}) is computed as [3, 57];

$$\tau_{dis-new} = K_a \cdot \mathrm{sgn}(S) + K_b.S \tag{49}$$

Based on Eq. (49) and Eq. (8);

$$\tau_{dis-PD-new} = K_a \cdot \mathrm{sgn}(\lambda e + \dot{e}) + K_b \cdot (\lambda e + \dot{e}) \tag{50}$$

According to Eq.(37) and Eq.(49);

$$\tau = \tau_{eq} + K_a \cdot \mathrm{sgn}(S) + +K_b - S$$
$$= \left[A^{-1}(q) \times \left(N(q, \dot{q}) \right) + \dot{S} \right] \times A(q) + K_a \cdot \mathrm{sgn}(S) + K_b \cdot S \tag{51}$$

According to Eq.(50) and Eq. (51) the formulation of PD-SMC is;

$$\tau_{PD-SMC-new} = K_a \cdot \mathrm{sgn}\left(\lambda e + \dot{e}\right) + K_b \cdot \left(\lambda e + \dot{e}\right)$$
$$+ \left[A^{-1}(q) \times \left(N(q, \dot{q}) \right) + \dot{S} \right] \times A(q) \tag{52}$$

Figure 2 shows the new chattering free PD sliding mode controller for serial links robot manipulator.

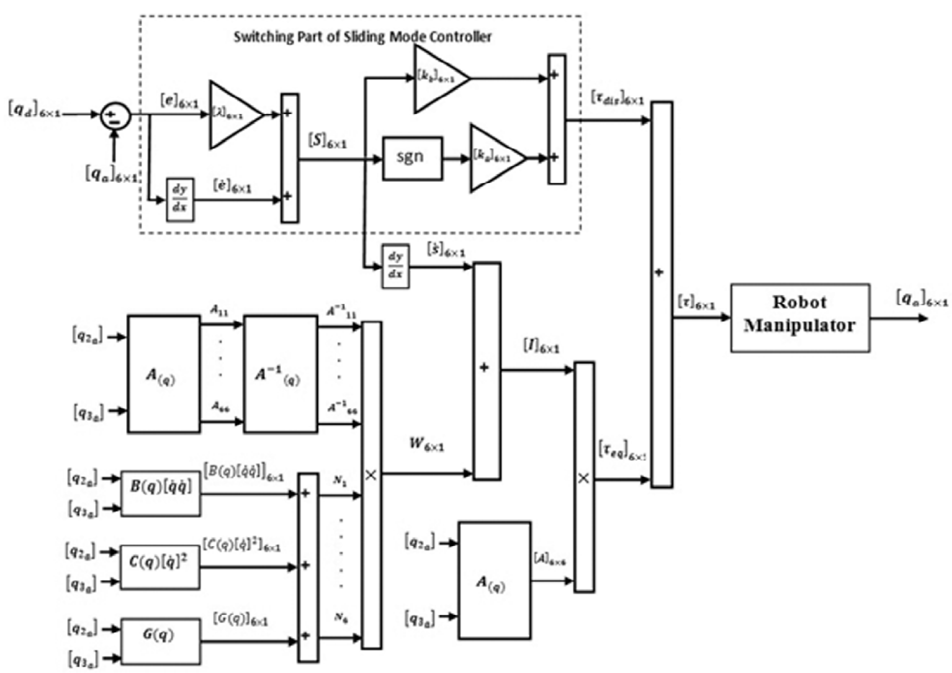

Figure 2: Block diagram of chatter-free Parallel Linear PD sliding mode controller for robot manipulator

Conventional sliding mode controller works based on manipulator dynamic model. Based on equivalent part in conventional nonlinear controllers in complex and highly nonlinear systems, these controllers encounter many problems for accurate responses because these type of controllers need to have accurate knowledge of dynamic formulation of the system. The nonlinear dynamic formulation problem in highly nonlinear system (e.g., robot manipulator) can be solved by means of fuzzy logic theorem. The system dynamics are estimated using Fuzzy logic theory. This type of controller has mathematical-free plant dynamic parameters. Despite of being used in many applications, pure fuzzy logic controllers have problems in pre sense of uncertainty condition (robust) and pre-define the inputs/outputs gain updating factors. To solve the equivalent challenge especially in uncertain system, fuzzy sliding mode controller employs the following formulation:

$$\tau_{eq}=\left[\hat{A}^{-1}(q)\times\left(\hat{N}(q,\dot{q})\right)+\dot{S}\right]\times\hat{A}(q) \tag{53}$$

When system works in uncertainty, the nonlinearity term of robot manipulator is not equal to equivalent term of sliding mode controller. To solve this challenge in this research PID like fuzzy logic controller is recommended as follows;

$$\tau_{eq-\ fuzzy\ estimate}=\left(\left[\hat{A}^{-1}(q)\times\left(\hat{N}(q,\dot{q})\right)+\dot{S}\right]\times\hat{A}(q)\right)+U_{PID-\ fuzzy} \tag{54}$$

Therefore the formulation of model-based PID like fuzzy sliding mode controller will be;

$$\tau_{PD-SMC-new}=K_a\cdot\operatorname{sgn}(\lambda e+\dot{e})+K_b\cdot(\lambda e+\dot{e})+\left(\left[\hat{A}^{-1}(q)\times\left(\hat{N}(q,\dot{q})\right)+\dot{S}\right]\times\hat{A}(q)\right) \tag{55}$$

PID like fuzzy logic controller has three inputs, Proportional (P), Derivative (D), and Integrator (I), if each input defined by N linguistic variables to estimate the dynamic behavior, it has $N\times N\times N$ linguistic variables. Designing fuzzy controller based on N^3 rule base for each link causes creation of lots of challenges in real time application. PID like fuzzy logic controller, parallel PD and PI strategy are recommended in order to reduce the number of rule base. According to this algorithm, designing the same PID controller will have $2N^2$ number of rule base for each link. This technique reduces the number of rule base as compared to PID like fuzzy logic controller. After solving the first challenge of reducing the number of rule base in PID like fuzzy logic controller, the second challenge appears. To design parallel PD and PI like fuzzy controller, two types of fuzzy rule table should be designed. Designing two types of rule tables are very difficult and requires a good deal of experience. Thus, in this research the PI-like fuzzy controller is replaced by PD-like fuzzy controller with the integral term in the output. As a result, researcher can design PID like fuzzy logic controller based on PD rule table and $2N^2$ rule base.

According to fuzzy logic methodology definition;

$$U_{fuzzy}=\left(\sum_{l=1}^{M}\theta^T\zeta(x)\right)_{e,\int e,\dot{e}} \tag{56}$$

where θ^T is gain updating factor and $\zeta(x)$ is defined by;

$$\zeta(x)=\frac{\sum_i\mu(x_i)x_i}{\sum_i\mu(x_i)} \tag{57}$$

And the $\mu(x_i)$ parameter is membership function.

A fuzzy logic controller is a nonlinear controller and this type of nonlinear controller does not require the dynamic model of nonlinear system to be controlled. Therefore this method can be applied to nonlinear system control (ie. robot manipulator) without determining the nonlinear dynamic model and solve the complicated equations. Figure 3 illustrates the general structure of the PID like fuzzy logic controller, which consists of two main components. The PID like fuzzy logic controller is built using PD like fuzzy logic controller and PI like fuzzy logic controller. PD like fuzzy controller and PI like fuzzy controller are designed based on PD fuzzy rule base. Thus, the PID like fuzzy controller is designed based on the following formulation;

$$U_{PID} = U_{PI} + U_{PD} = \left(\frac{K_p}{2}\right) \times e + K_i \left(\frac{1}{T}\int e.dt\right) + \left(\frac{K_p}{2}\right) \times e + K_v \dot{e} \tag{58}$$

$$U_{PID\ like\ fuzzy} = \left[\left(\sum_{l=1}^{M} \theta^T \zeta(x)\right)_{e,\int e} + \left(\sum_{l=1}^{M} \theta^T \zeta(x)\right)_{e,\dot{e}}\right] \tag{59}$$

PID like fuzzy logic controller consists of the following parts;

- Choosing inputs
- Scaling inputs
- Input fuzzification (binary-to-fuzzy[B/F] conversion)
- Fuzzy rule base (knowledge base)
- Inference engine
- Output defuzzification (fuzzy-to-binary[F/B] conversion)
- Scaling output

Define the inputs and control variables: In PID-like fuzzy controller, error and change of error are used to define the controllers' inputs. Therefore the antecedent part of rule base is divided into two parts. These

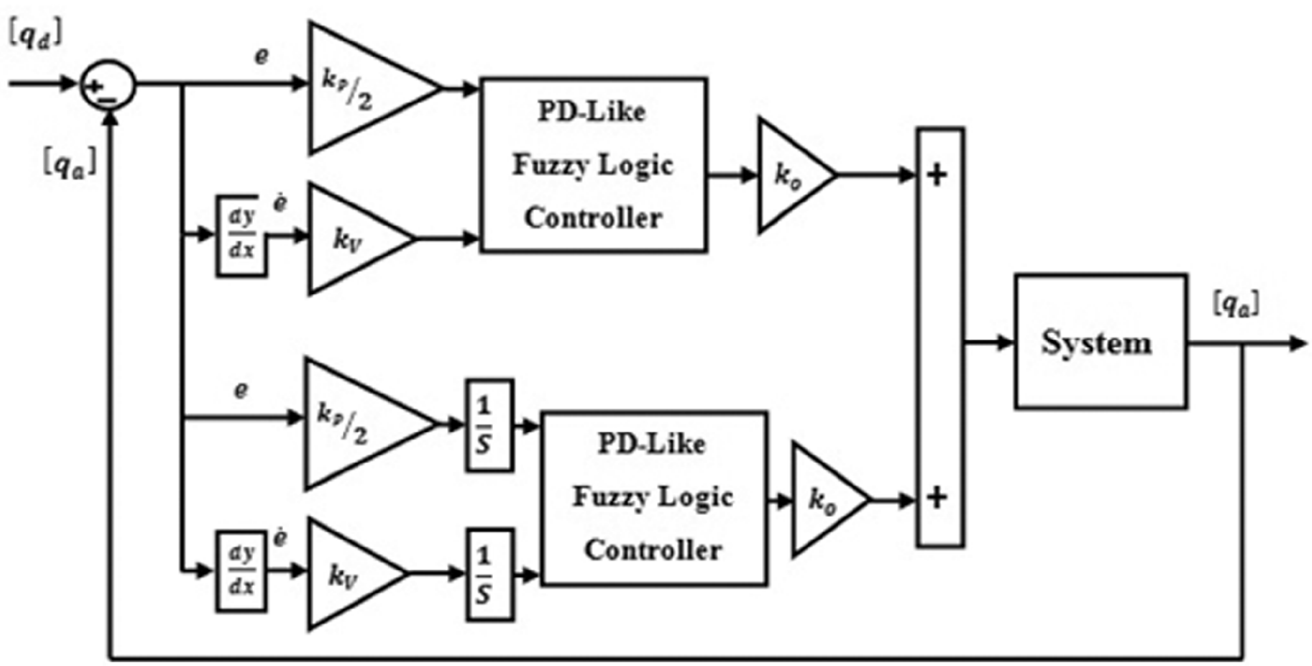

Figure 3: Design of PID like fuzzy controller

parts are the fuzzy controller inputs which consists of error (e) and change of error ($\dot{e}$) and the fuzzy controller outputs which comprises the PD fuzzy output ($U_{PD-fuzzy}$) and PI fuzzy output ($U_{PI-fuzzy}$).

Scaling variables and Input fuzzification (binary-to-fuzzy [B/F] conversion):

The proposed PID like fuzzy logic controller has two inputs, ie. *error* and *change of error* and two different output, ie. *PD fuzzy output and PI fuzzy output.* Error is defined as seven linguistic variables: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB). Based on experience, the range of scaling factor for error is [–0.1 *to* 0.1] and it is quantized into eleven levels as follow: e = {0.1, –0.08, –0.06, –0.04, –0.02, 0, 0.02, 0.04, 0.06, 0.08, 0.1}. The linguistic values for change of error are: Negative (N), Zero (Z) and Positive (P) and the range of scaling factor for change of error is [–1 *to* 1] and it is quantized into eleven levels as: $\dot{e}$ = {–1, –0.8, –0.6, –0.4, –0.2, 0, 0.2, 0.4, 0.6, 0.8, 1}. The linguistic variables for PID like fuzzy logic controller's output are divided into two main parts, PD like fuzzy logic controller and PI like fuzzy logic controller which, the linguistic variables for output PD like fuzzy logic controller and PI like fuzzy logic controller are: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB) and the scaling factor for them are [–1.5 *to* 1.5].

Fuzzy rule Base: The main approach comes from an expert knowledge of system, where fuzzy controller is one of the expert system that is able to solve the control problem. According to fuzzification, the error has seven linguistic variables, the change of error has three linguistic variables and the PD fuzzy output and PI like fuzzy logic controller have seven linguistic variables. Therefore PID like fuzzy controller has 42 rule-bases. The PID like fuzzy rule table is shown in Table 1.

Table 1
Rule table in PID like fuzzy logic controller

$\dot{e}$ \ e	*PB*	*PM*	*PS*	*Z*	*NS*	*NM*	*NB*
P	NB	NB	NB	NB	NM	NS	Z
Z	NB	NM	NS	Z	PS	PM	PB
N	Z	PS	PM	PB	PB	PB	PB

Inference Engine (Fuzzy rule processing): In this research 42 rule base Mamdani fuzzy inference engine is used as fuzzy rule processing.

Defuzzification: Defuzzification is the last step to design fuzzy logic controller and it is used to transform fuzzy set to crisp set. In PID like fuzzy logic controller COG method is used for defuzzification.

According to the dynamic formulation of robot manipulators, they are uncertain and there are strong coupling effects between joints. Conventional sliding mode controller works based on manipulator dynamic model. This controller has two important subparts, which are namely the switching part and equivalent part. Equivalent part of sliding mode controller is used to eliminate the decoupling and nonlinear term of dynamic parameters of each link. Even though the equivalent part is very essential to reliability in uncertain condition or highly nonlinear dynamic systems it can cause some problems. Therefore, to solve this challenge the PID fuzzy logic controller is used as a parallel controller with sliding mode controller as a model-based PID like fuzzy sliding mode controller. To solve the number of rule base in fuzzy model-base sliding mode controller, parallel strategy is evaluated based on parallel structure of the PD-like fuzzy controller and PI-like fuzzy controller. Both PD and PI fuzzy controller are extracted from PD rule table. Therefore PI like fuzzy controller is replaced by PD-like fuzzy controller with the integral term in the output. Therefore proposed model-based PID like fuzzy sliding mode controller is recommended based on PD fuzzy rule table with reduced number of rule base.

Based on conventional sliding mode controller;

$$\tau_{PD-SMC-new} = \tau_{Dis-New} + \tau_{eq} \tag{60}$$

Therefore $\tau_{Dis-New}$ is calculated by;

$$\tau_{Dis-new} = K_a \cdot \text{sgn}(\lambda e + \dot{e}) + K_b \cdot (\lambda e + \dot{e}) \tag{61}$$

And τ_{eq} is obtained by;

$$\tau_{eq} = \left[\hat{A}^{-1}(q) \times \left(\widehat{N}(q,\dot{q})\right) + \dot{S}\right] \times \hat{A}(q) \tag{62}$$

PID like fuzzy sliding mode controller is recommended to solve the equivalent challenge especially in uncertain system. The formulation of PID like fuzzy sliding mode controller is;

$$\tau_{PID\ like\ fuzzy\text{-}new} = \tau_{Dis\text{-}New} + (\tau_{eq} + \tau_{PID\ like\ fuzzy}) \tag{63}$$

In $\tau_{PID\ like\ fuzzy,}$ inputs are error (e) and change of error ($\dot{e}$) while the fuzzy controller output is PD fuzzy output ($U_{PD\text{-}fuzzy}$) and PI fuzzy output ($U_{PI\text{-}fuzzy}$).

$$U_{PID\ like\ fuzzy} = U_{PI\ like\ fuzzy} + U_{PD\ like\ fuzzy} = \left[\left(\Sigma_{l=1}^{M} \theta^T \zeta(x)\right)_{e\int e,} + \left(\Sigma_{l=1}^{M} \theta^T \zeta(x)\right)_{e,\dot{e}}\right] \tag{64}$$

According to the above formulation;

$$\left(\tau_{eq} + \tau_{PIDlikefuzzy}\right) = \left[\hat{A}^{-1}(q) \times \left(\widehat{N}(q,\dot{q})\right) + \dot{S}\right] \times$$
$$\hat{A}(q) + \left[\left(\Sigma_{l=1}^{M} \theta^T \zeta(x)\right)_{e\int e} + \left(\Sigma_{l=1}^{M} \theta^T \zeta(x)\right)_{e,\dot{e}}\right] \tag{65}$$

$$\tau_{PID\ like\ fuzzy\ SMC-new} = K_a \cdot \text{sgn}\left(\lambda e + \dot{e}\right) + K_b \cdot \left(\lambda e + \dot{e}\right) +$$
$$\left(\left[\hat{A}^{-1}(q) \times \left(\widehat{N}(q,\dot{q})\right) + \dot{S}\right] \times \hat{A}(q) + \hat{A}(q) + \left[\left(\Sigma_{l=1}^{M} \theta^T \zeta(x)\right)_{e\int e} + \left(\Sigma_{l=1}^{M} \theta^T \zeta(x)\right)_{e,\dot{e}}\right]\right) \tag{66}$$

The above algorithm describes that to design the same PID controller the number of rule base required for each link is $2 \times N \times M$. Where N is the number of linguistic variables for error and M is the number of linguistic variables for change of error. According to this technique, the number of rule base is reduced with respect to PID like fuzzy logic controller. In this research N is equal to seven variables namely the Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), Positive Big (PB) and M is three variables, Negative (N), Zero (Z) and Positive (P).

Figure 4 illustrates the PID like fuzzy sliding mode controller.The PID like fuzzy sliding mode controller can be updated based on online tuning sliding surface slope. In order to reduce the online computation burden, the PID like fuzzy logic controller is also used in sliding surface slope online tuning. Figure 5 illustrates the structure of online tuning of sliding surface slope PID like fuzzy sliding mode controller.

Referring to Figure 5, PID like fuzzy logic controller is used to increase the system stability in presence of system uncertainty.

$$S_{PD} = \lambda e + \dot{e} \tag{67}$$

$$\lambda_{update} = \lambda \times \tau_{PID\ like\ fuzzy} \tag{68}$$

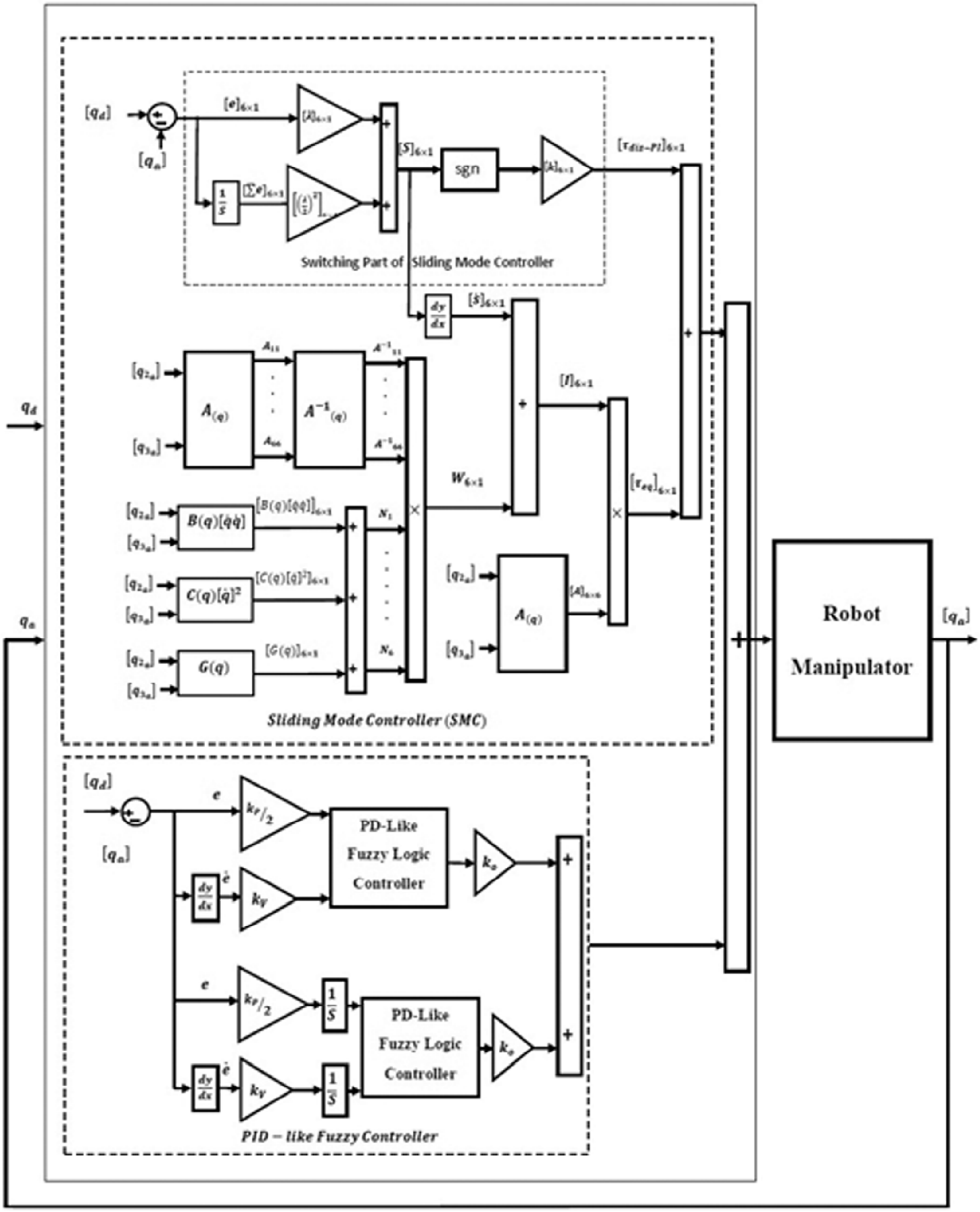

Figure 4:Design PID like fuzzy sliding mode controller

$$\lambda_{update} = \lambda \times \left[\left(\sum_{l=1}^{M} \theta^T \zeta(x) \right)_{e, \int e} + \left(\sum_{l=1}^{M} \theta^T \zeta(x) \right) e, \dot{e} \right] \tag{69}$$

Where λ_{update} is online tuning of sliding surface slope using PID like fuzzy logic controller. According to Eq. (68) the modified online tuning sliding surface slope PID like fuzzy sliding mode controller is;

$$\tau_{Modify\ PID\ like\ fuzzy\ SMC-new} = K \cdot \mathrm{sgn}\left(\lambda_{update} e + \dot{e}\right) + K_b \cdot \left(\lambda_{update} e + \dot{e}\right) +$$

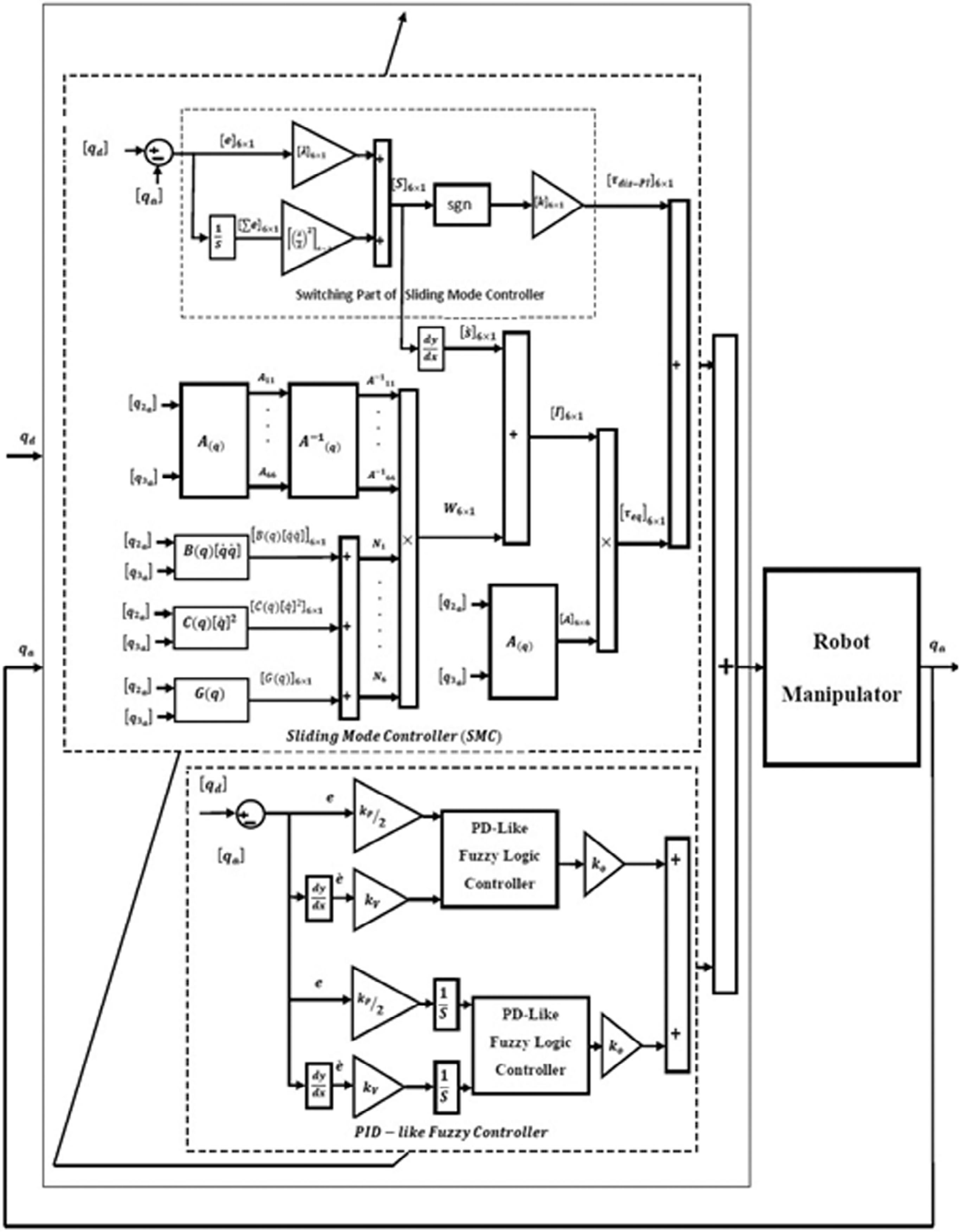

Figure 5: Design online tuning sliding surface slope PID like fuzzy sliding mode controller

$$\left(\left[\hat{A}^{-1}(q)\times\left(\hat{N}(q,\dot{q})\right)+\dot{S}\right]\times\hat{A}(q)+\left[\left(\sum_{l=1}^{M}\theta^T\zeta(x)\right)_{e,\int e}+\left(\sum_{l=1}^{M}\theta^T\zeta(x)\right)_{e,\dot{e}}\right]\right) \quad (70)$$

4. RESULTS AND DISCUSSIONS

In this section, a robot manipulator is used as a benchmark model to evaluate the control algorithms. The following managements are being compared: online sliding surface slope tuning PID like fuzzy sliding mode controller and offline sliding surface slope tuning PID like fuzzy sliding mode controller. Both controllers are applied to a 6-DOF serial links robot. The simulation was implemented in MATLAB/ SIMULINK environment.

Comparison of the Tracking Data and Information: Based on the formulation of PID like fuzzy sliding mode controller formulation, discontinuous controllers gain (K_a) linear controllers gain (K_b), PD gain updating factors ($K_{p\,PD}$, K_v and $K_{o\,PD}$), PI gain updating factors ($K_{p\,PI}$, K_I and $K_{o\,PI}$) and sliding surface slope (λ) are giving significant impact on the system's performance. Sliding surface slope is the main coefficient to design conventional sliding mode controller, parallel linear chattering free sliding mode controller and PID like fuzzy sliding mode controller. Hence, to improve the controller's performance as well as to increase the controller robustness, online tuning sliding surface slope is recommended. In uncertain situations, sliding surface slope can perform online tuning by means of PID like fuzzy logic controller. According to this theory, the performance of online tuning is better than offline tuning PID like fuzzy sliding mode controller.

The trajectory following of 6 DOF for online sliding surface slope tuning PID like fuzzy sliding mode controller and off line tuning surface slope tuning PID like fuzzy sliding mode controller are compared in Figure 6. Based on Figure 6, both controllers can eliminate the chattering and oscillation in certain situation. In rise time point of view, offline PID like fuzzy sliding mode controller is faster than online PID like fuzzy sliding mode controller because the rise time in offline PID like fuzzy sliding mode controller is 0.48

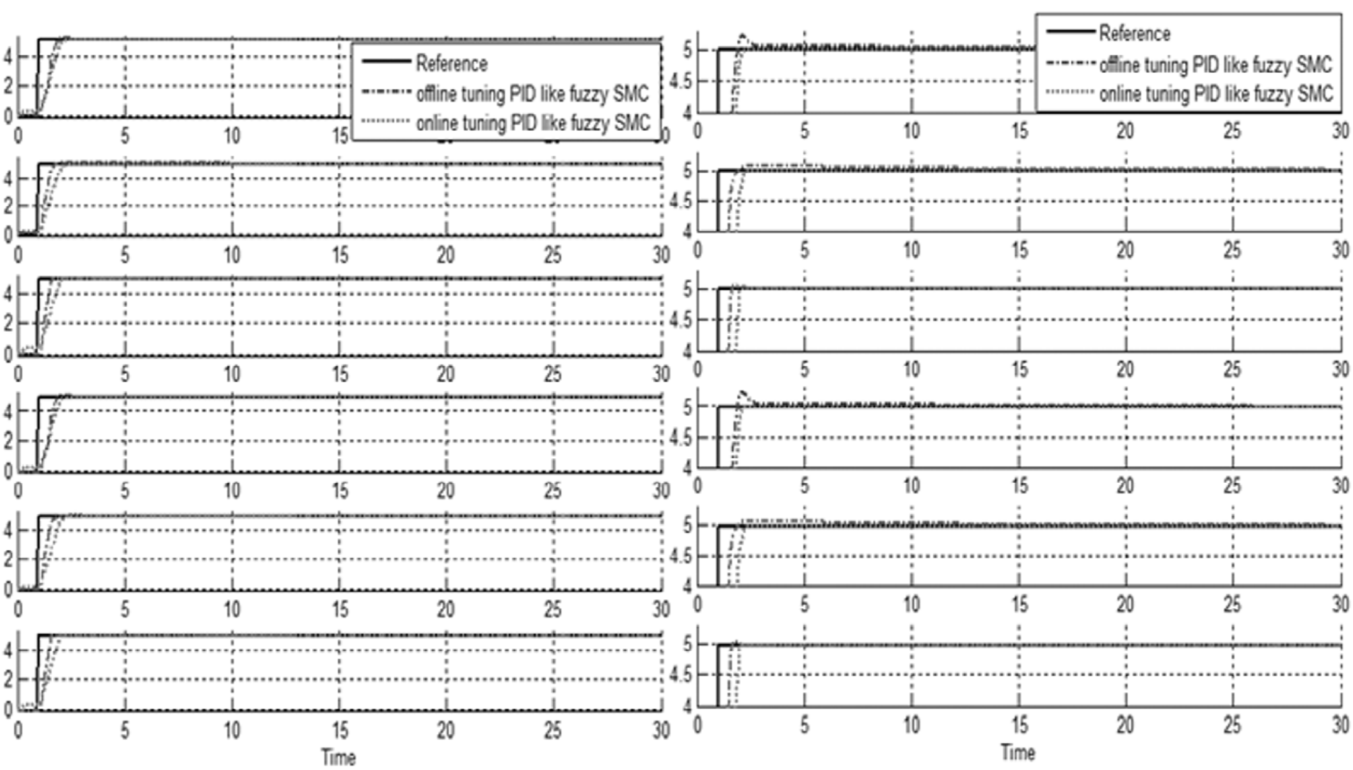

Figure 6: Trajectory following: Offline PID like fuzzy SMC and online PID like fuzzy SMC

second while in online PID like fuzzy sliding mode controller is 0.50 second. In error point of view, online PID like fuzzy sliding mode controller is better than offline PID like fuzzy sliding mode controller. According to Figure 6, online PID like fuzzy sliding mode controller has accurate trajectory response and it can eliminate the chattering as well as reduce the error.

Comparison of sliding surface (*S*): Figure 7 shows the sliding surface in offline sliding surface slope tuning PID like fuzzy sliding mode controller and online sliding surface slope tuning PID like fuzzy sliding mode controller. According to the following graphs, both methods have about the same sliding surface trajectories and these trajectories are zero.

Based on Figure 7, sliding surface of offline sliding surface slope tuning PID like fuzzy sliding mode controller and online sliding surface slope tuning PID like fuzzy sliding mode controller are spike free, which prove the stability.

Comparison of the actuation torque (τ_i): The control input, forces the robot manipulators to track the desired trajectories. Figure 8 shows the torque performance in offline sliding surface slope tuning PID like fuzzy sliding mode controller and online sliding surface slope tuning PID like fuzzy sliding mode controller. According to the following graphs, both controllers have steady and stable torque performance.

Referring to Figure 8, the amplitude of the control forces in conventional sliding mode controller is much larger than offline and online sliding surface slope tuning PID like fuzzy sliding mode controllers. In the control forces, smaller amplitude means less energy. Therefore, offline and online sliding surface slope tuning PID like fuzzy sliding mode controllers require less energy than the conventional sliding mode controller.

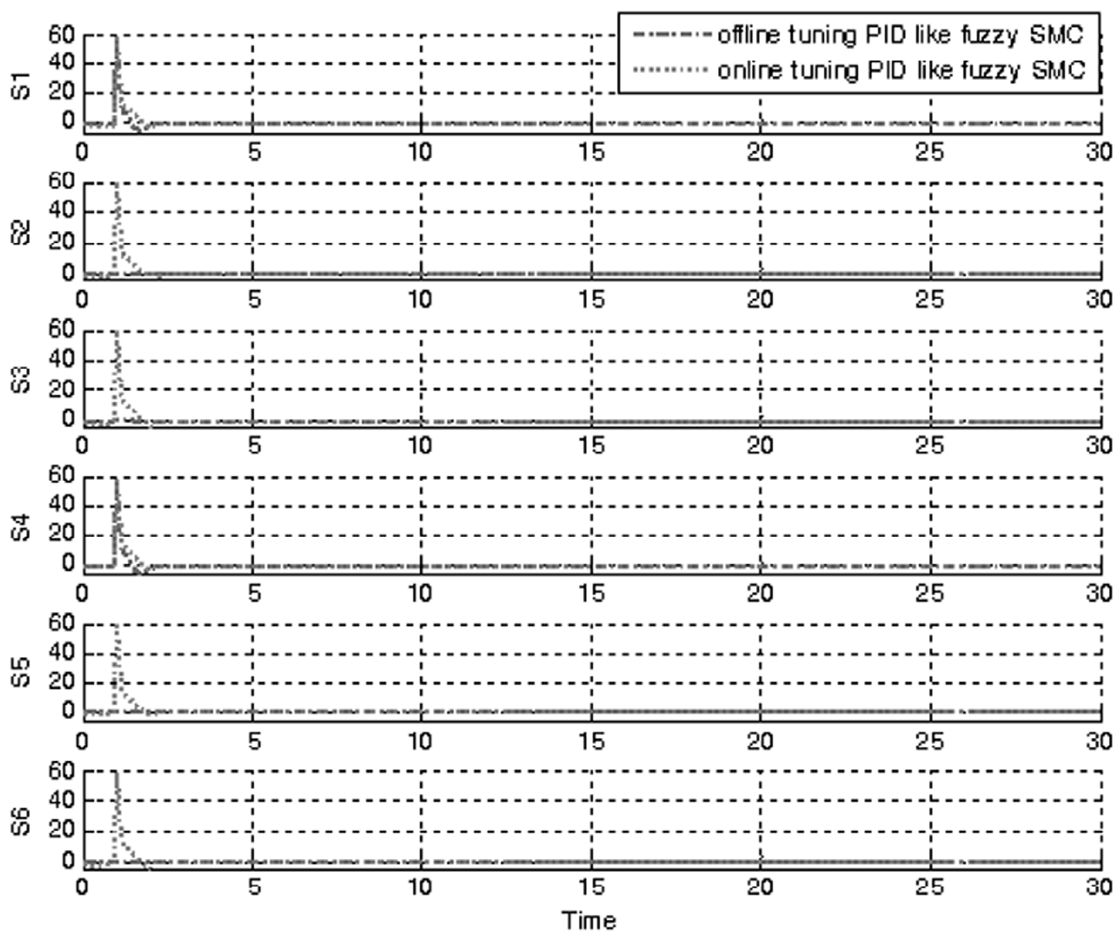

Figure 7: Comparison of sliding surface: Offline PID like fuzzy SMC and online PID like fuzzy SMC

Comparison of disturbance rejection: The power of disturbance rejection is very important to robust checking in these two types of controllers. In this section the trajectory accuracy, sliding surface and torque performances are tested under uncertain condition. A limited white noise with 30% amplitude is applied to offline sliding surface slope tuning PID like fuzzy sliding mode controller and online sliding surface slope tuning PID like fuzzy sliding mode controller in order to test the disturbance rejection band. The trajectory accuracy, sliding surface and torque performance are shown in Figures 9 to 11.

According to the above graphs, online tuning sliding surface slope tuning PID like fuzzy sliding mode controller is more stable as compared to online tuning sliding surface slope tuning PID like fuzzy sliding mode controller. In the presence of uncertainty, online tuning estimates the sliding surface slope using PID like fuzzy logic controller. Whereas offline sliding surface slope tuning PID like fuzzy sliding mode controller has moderate fluctuations in the presence of uncertainty. Figure 10 shows the sliding surface in the presence of uncertainty.

The above graphs prove that, even though offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller can eliminate the chattering, it still has fluctuations during the presence of uncertainties. This is the main challenge in offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller, where sliding surface slope cannot adjust the sliding surface in the presence of uncertainty. Again, referring to the above graph, online tuning sliding surface slope tuning PID like fuzzy sliding mode

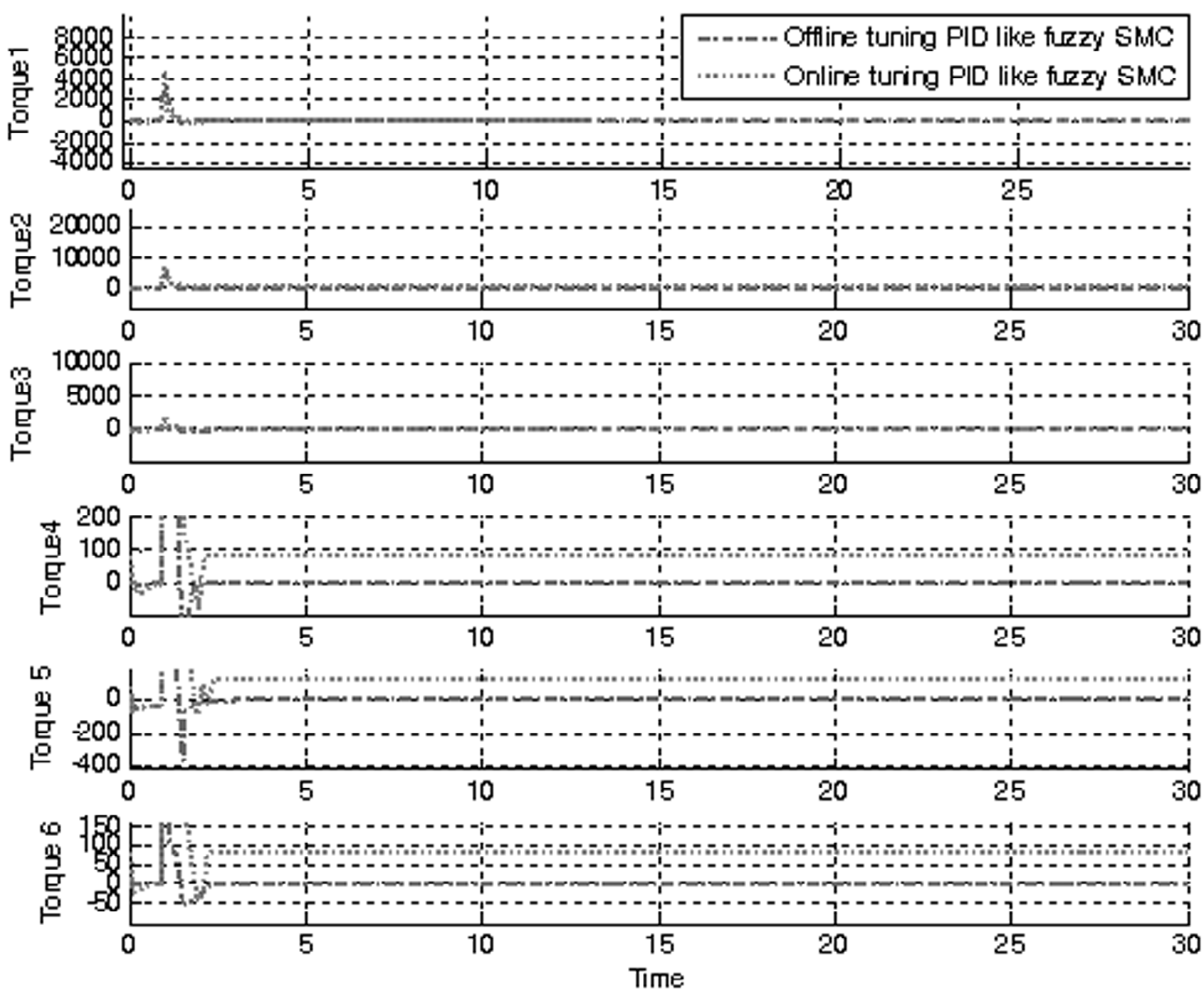

Figure 8: Comparison of the actuation torque-Offline PID like fuzzy SMC and online PID like fuzzy SMC

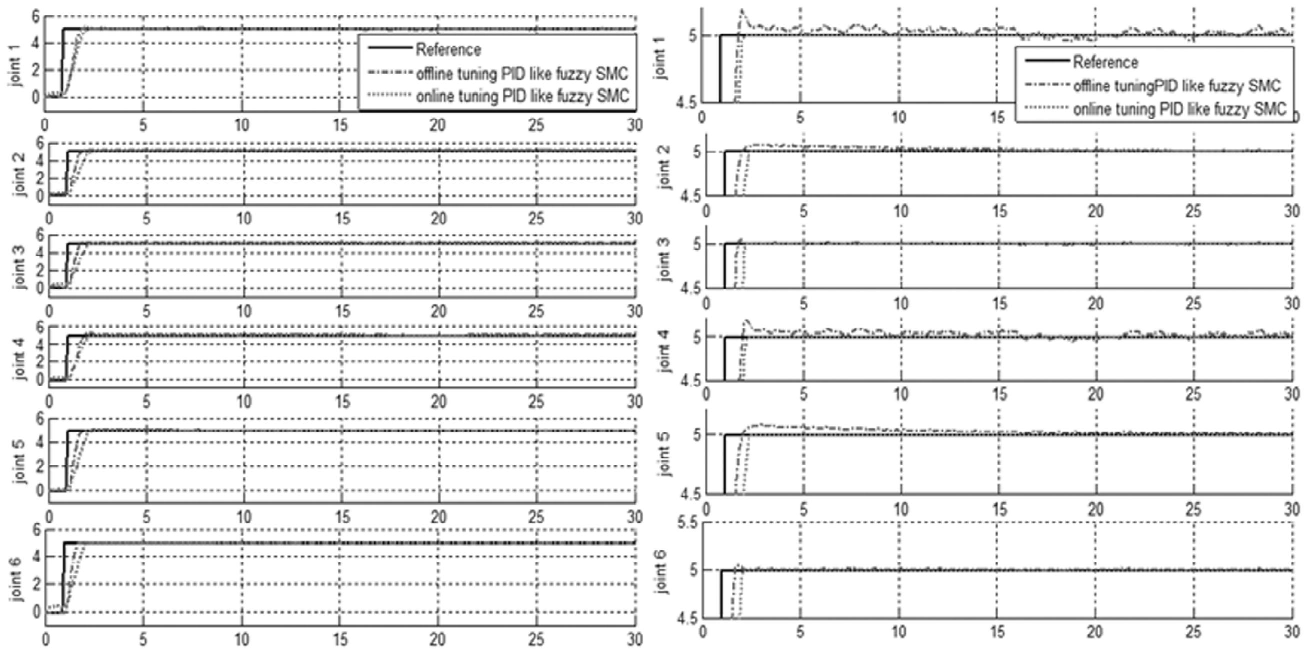

Figure 9: Comparison of disturbance rejection: Offline PID like fuzzy SMC and online PID like fuzzy SMC in presence of uncertainty

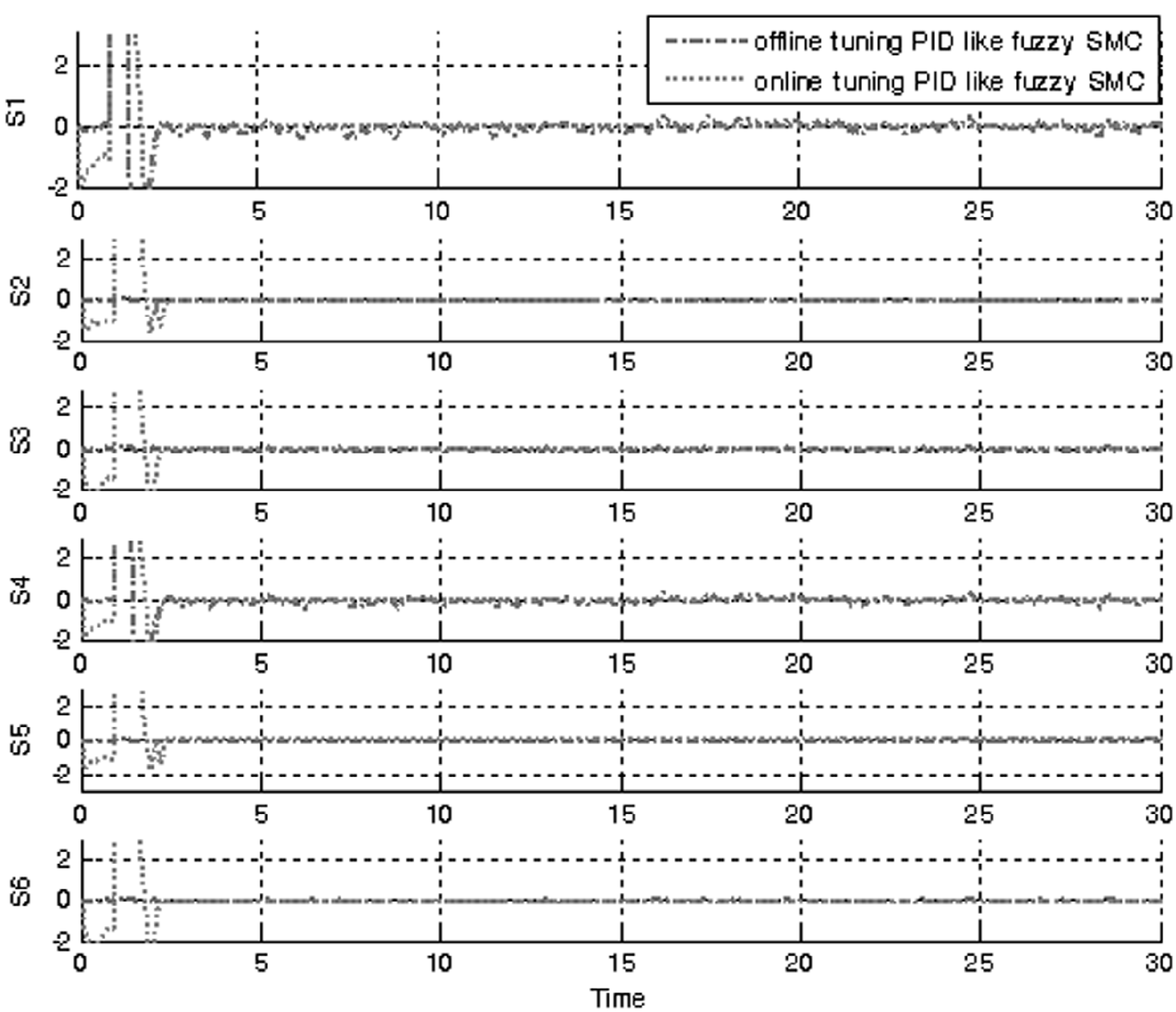

Figure 10: Comparison of sliding surface: Offline PID like fuzzy SMC and online PID like fuzzy SMC in the presence of uncertainty

controller is more robust than the offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller, where the amplitude of fluctuation is near to zero. Figure 11 shows the torque performance in the presence of uncertainty.

Based on the above graphs, offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller has moderate oscillation in the presence of uncertainty. According to the above three graphs, online tuning sliding surface slope tuning PID like fuzzy sliding mode controller is more stable than the offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller because it has online tunable gain.

Tracking error comparison: This part is used to test the controller joint variable accuracy. Figure 12 shows the steady state error in online tuning sliding surface slope tuning PID like fuzzy sliding mode controller and offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller. According to this Figure, offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller has irregular fluctuations but online tuning sliding surface slope tuning PID like fuzzy sliding mode controller has steady stability.

According to Figure 12 above, even though the online tuning sliding surface slope tuning PID like fuzzy sliding mode controller and offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller have about the same error trajectory, but the online tuning sliding surface slope tuning PID like

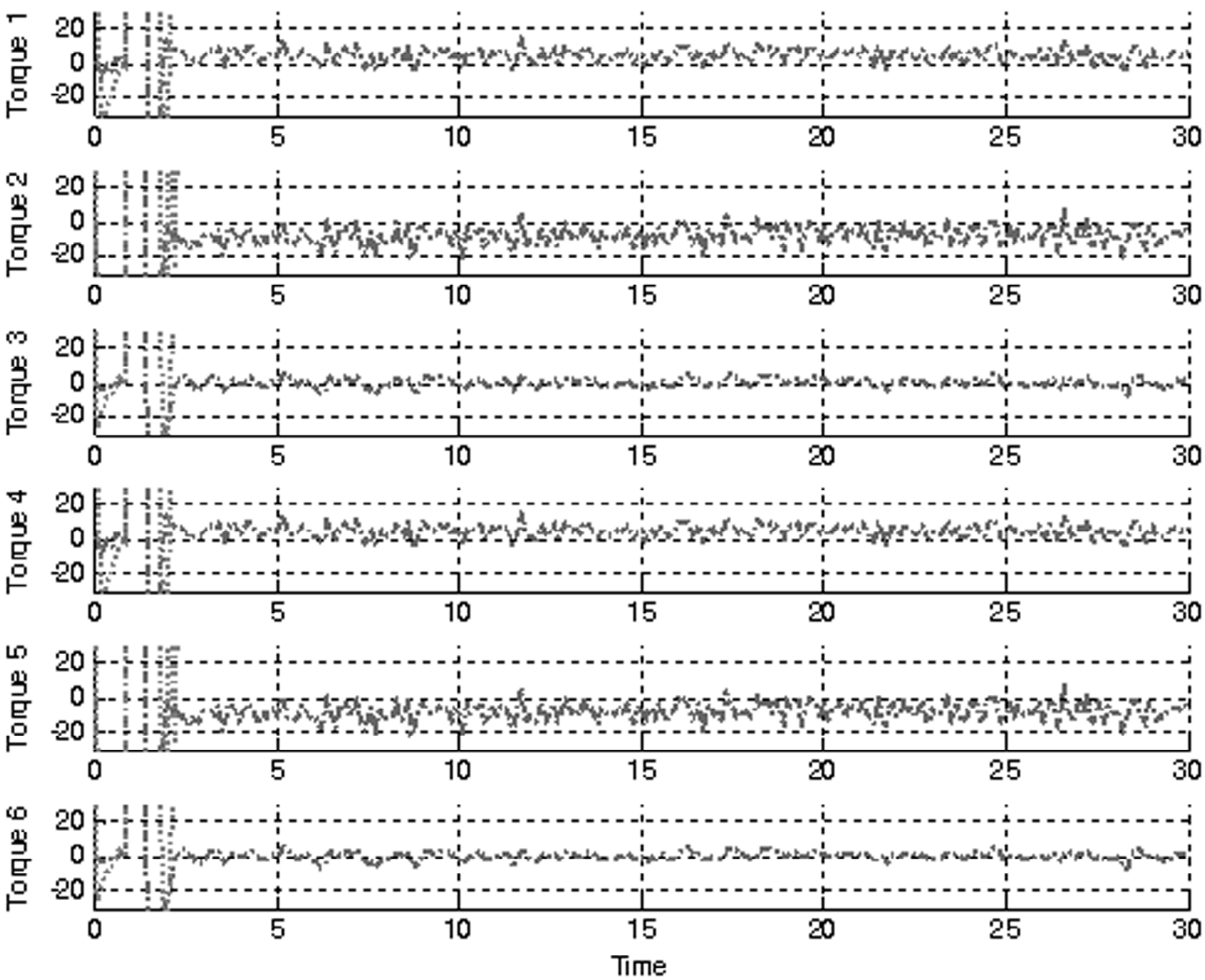

Figure 11: Comparison of the actuation torque:
Offline PID like fuzzy SMC and online PID like fuzzy SMC in the presence of uncertainty

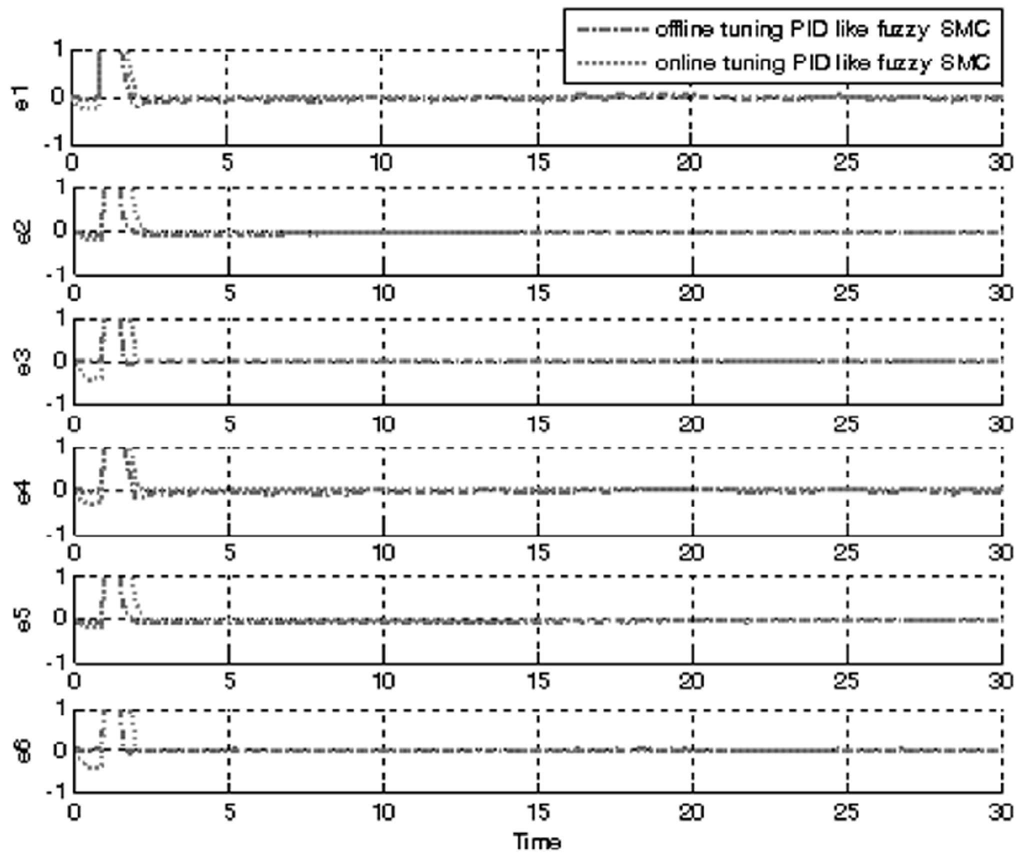

Figure 12: Comparison of the steady state error:
Offline PID like fuzzy SMC and online PID like fuzzy SMC in the presence of uncertainty

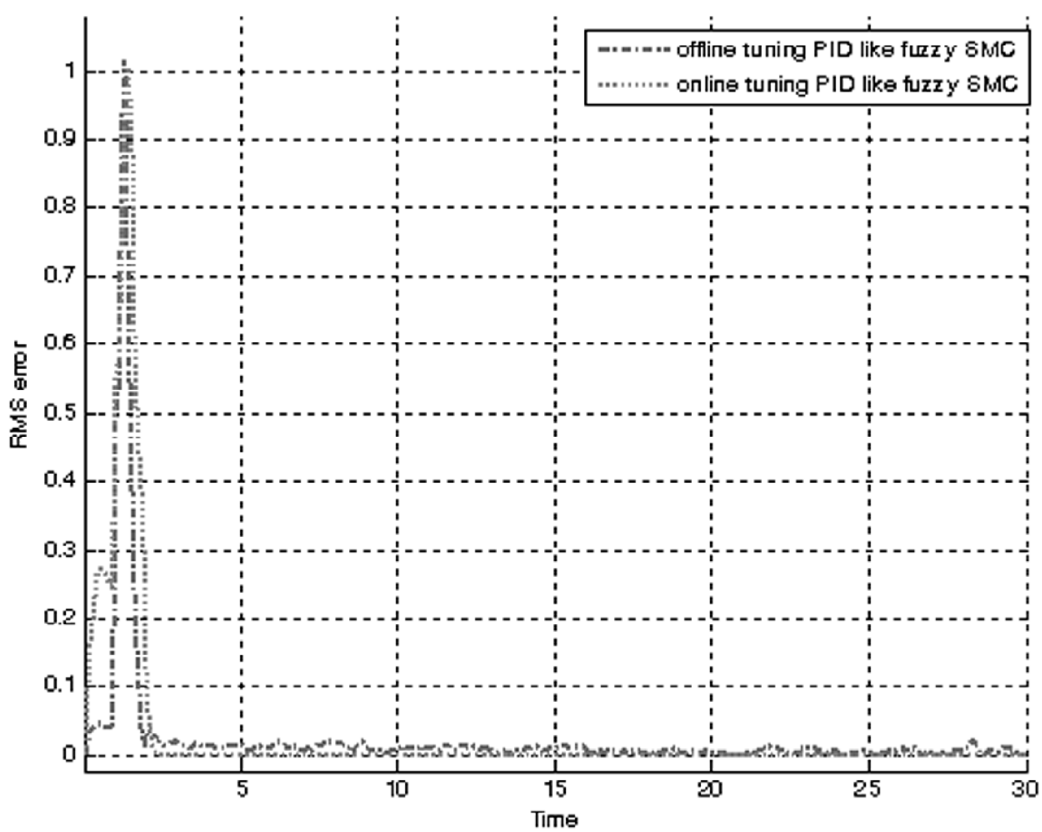

Figure 13: Comparison of the RMS error:
Offline PID like fuzzy SMC and online PID like fuzzy SMC in presence of uncertainty

fuzzy sliding mode controller is found to be more robust than offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller.

Figure 13 shows root means square (RMS) error in the presence of uncertainty for online tuning sliding surface slope tuning PID like fuzzy sliding mode controller and offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller.Based on Figure 13, offline tuning sliding surface slope tuning PID like fuzzy sliding mode controller has more position deviation than online tuning sliding surface slope tuning PID like fuzzy sliding mode controller.

5. CONCLUSIONS

According to the dynamic formulation of robot manipulators, they are uncertain and have strong coupling effects between joints. To solve this challenge, online sliding surface slope tuning PID like fuzzy sliding mode controller is selected because this type of controller is robust, stable and works very well in certain and uncertain situations. In the proposed method, the chattering effect as well as the steady state error can be eliminated by adding the linear control theory to discontinuous part. While to reduce the number of rules in fuzzy logic method, PD like fuzzy controller plus PI like fuzzy controller is used as a PID controller. In this research, the sliding surface slopes are tuned online based on PID like fuzzy logic controller, and fuzzy logic controller is used in dynamic estimation and also online tuning. In the N degrees of freedom robot manipulator, if K membership functions are defined for each input variable, the number of fuzzy rules for each joint of robot manipulator is $2K^2$ which means it is obviously decreased.

REFERENCES

[1] T. R. Kurfess, Robotics and automation handbook: CRC, 2005.

[2] J. J. E. Slotine and W. Li, Applied nonlinear control vol. 461: Prentice hall Englewood Cliffs, NJ, 1991.

[3] L. Cheng, et al., “Multi-agent based adaptive consensus control for multiple manipulators with kinematic uncertainties,” 2008, pp. 189-194.

[4] J. J. D’Azzo, et al., Linear control system analysis and design with MATLAB: CRC, 2003.

[5] B. Siciliano and O. Khatib, Springer handbook of robotics: Springer-Verlag New York Inc, 2008.

[6] I. Boiko, et al., “Analysis of chattering in systems with second-order sliding modes,” IEEE Transactions on Automatic Control, vol. 52, pp. 2085-2102, 2007.

[7] J. Wang, et al., “Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching,” Fuzzy Sets and Systems, vol. 122, pp. 21-30, 2001.

[8] F. Piltan, et al., “Artificial Control of Nonlinear Second Order Systems Based on AFGSMC,” Australian Journal of Basic and Applied Sciences, 5(6), pp. 509-522, 2011.

[9] V. Utkin, “Variable structure systems with sliding modes,” Automatic Control, IEEE Transactions on, vol. 22, pp. 212-222, 2002.

[10] R. A. DeCarlo, et al., “Variable structure control of nonlinear multivariable systems: a tutorial,” Proceedings of the IEEE, vol. 76, pp. 212-232, 2002.

[11] K. D. Young, et al., “A control engineer’s guide to sliding mode control,” 2002, pp. 1-14.

[12] O. Kaynak, “Guest editorial special section on computationally intelligent methodologies and sliding-mode control,” IEEE Transactions on Industrial Electronics, vol. 48, pp. 2-3, 2001.

[13] J. J. Slotine and S. Sastry, “Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators†,” International Journal of Control, vol. 38, pp. 465-492, 1983.

[14] J. J. E. Slotine, “Sliding controller design for non-linear systems,” International Journal of Control, vol. 40, pp. 421-434, 1984.

[15] R. Palm, “Sliding mode fuzzy control,” 2002, pp. 519-526.

[16] C. C. Weng and W. S. Yu, “Adaptive fuzzy sliding mode control for linear time-varying uncertain systems,” 2008, pp. 1483-1490.

[17] M. Ertugrul and O. Kaynak, “Neuro sliding mode control of robotic manipulators,” Mechatronics, vol. 10, pp. 239-263, 2000.

[18] P. Kachroo and M. Tomizuka, “Chattering reduction and error convergence in the sliding-mode control of a class of nonlinear systems,” Automatic Control, IEEE Transactions on, vol. 41, pp. 1063-1068, 2002.

[19] H. Elmali and N. Olgac, “Implementation of sliding mode control with perturbation estimation (SMCPE),” Control Systems Technology, IEEE Transactions on, vol. 4, pp. 79-85, 2002.

[20] J. Moura and N. Olgac, “A comparative study on simulations vs. experiments of SMCPE,” 2002, pp. 996-1000.

[21] Y. Li and Q. Xu, “Adaptive Sliding Mode Control With Perturbation Estimation and PID Sliding Surface for Motion Tracking of a Piezo-Driven Micromanipulator,” Control Systems Technology, IEEE Transactions on, vol. 18, pp. 798-810, 2010.

[22] B. Wu, et al., “An integral variable structure controller with fuzzy tuning design for electro-hydraulic driving Stewart platform,” 2006, pp. 5-945.

[23] L. A. Zadeh, “Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic,” Fuzzy Sets and Systems, vol. 90, pp. 111-127, 1997.

[24] L. Reznik, Fuzzy controllers: Butterworth-Heinemann, 1997.

[25] J. Zhou and P. Coiffet, “Fuzzy control of robots,” 2002, pp. 1357-1364.

[26] S. Banerjee and P. Y. Woo, “Fuzzy logic control of robot manipulator,” 2002, pp. 87-88.

[27] K. Kumbla, et al., “Soft computing for autonomous robotic systems,” Computers and Electrical Engineering, vol. 26, pp. 5-32, 2000.

[28] C. C. Lee, “Fuzzy logic in control systems: fuzzy logic controller. I,” IEEE Transactions on systems, man and cybernetics, vol. 20, pp. 404-418, 1990.

[29] R. J. Wai, et al., “Implementation of artificial intelligent control in single-link flexible robot arm,” 2003, pp. 1270-1275.

[30] R. J. Wai and M. C. Lee, “Intelligent optimal control of single-link flexible robot arm,” Industrial Electronics, IEEE Transactions on, vol. 51, pp. 201-220, 2004.

[31] M. B. Menhaj and M. Rouhani, “A novel neuro-based model reference adaptive control for a two link robot arm,” 2002, pp. 47-52.

[32] S. Mohan and S. Bhanot, “Comparative study of some adaptive fuzzy algorithms for manipulator control,” International Journal of Computational Intelligence, vol. 3, pp. 303–311, 2006.

[33] F. Barrero, et al., “Speed control of induction motors using a novel fuzzy sliding-mode structure,” Fuzzy Systems, IEEE Transactions on, vol. 10, pp. 375-383, 2002.

[34] Y. C. Hsu and H. A. Malki, “Fuzzy variable structure control for MIMO systems,” 2002, pp. 280-285.

[35] Y. C. Hsueh, et al., “Self-tuning sliding mode controller design for a class of nonlinear control systems,” 2009, pp. 2337-2342.

[36] R. Shahnazi, et al., “Position control of induction and DC servomotors: a novel adaptive fuzzy PI sliding mode control,” Energy Conversion, IEEE Transactions on, vol. 23, pp. 138-147, 2008.

[37] C. C. Chiang and C. H. Wu, “Observer-Based Adaptive Fuzzy Sliding Mode Control of Uncertain Multiple-Input Multiple-Output Nonlinear Systems,” 2007, pp. 1-6.

[38] H. Temeltas, “A fuzzy adaptation technique for sliding mode controllers,” 2002, pp. 110-115.

[39] C. L. Hwang and S. F. Chao, “A fuzzy-model-based variable structure control for robot arms: theory and experiments,” 2005, pp. 5252-5258.

[40] C. G. Lhee, et al., “Sliding mode-like fuzzy logic control with self-tuning the dead zone parameters,” Fuzzy Systems, IEEE Transactions on, vol. 9, pp. 343-348, 2002.

[41] Lhee. C. G., J. S. Park, H. S. Ahn, and D. H. Kim, “Sliding-Like Fuzzy Logic Control with Self-tuning the Dead Zone Parameters,” IEEE International fuzzy systems conference proceeding, 1999, pp. 544-549.

[42] X. Zhang, et al., “Adaptive sliding mode-like fuzzy logic control for high order nonlinear systems,” pp. 788-792.

[43] M. R. Emami, et al., “Development of a systematic methodology of fuzzy logic modeling,” IEEE Transactions on Fuzzy Systems, vol. 6, 1998.

[44] H.K.Lee, K.Fms, “A Study on the Design of Self-Tuning Sliding Mode Fuzzy Controller. Domestic conference,” IEEE Conference, 1994, vol. 4, pp. 212-218.

[45] Z. Kovacic and S. Bogdan, Fuzzy controller design: theory and applications: CRC/Taylor & Francis, 2006.

[46] F. Y. Hsu and L. C. Fu, "Nonlinear control of robot manipulators using adaptive fuzzy sliding mode control," 2002, pp. 156-161.

[47] R. G. Berstecher, et al., "An adaptive fuzzy sliding-mode controller," Industrial Electronics, IEEE Transactions on, vol. 48, pp. 18-31, 2002.

[48] V. Kim, "Independent joint adaptive fuzzy control of robot manipulator," 2002, pp. 645-652.

[49] Y. Wang and T. Chai, "Robust adaptive fuzzy observer design in robot arms," 2005, pp. 857-862.

[50] B. K. Yoo and W. C. Ham, "Adaptive control of robot manipulator using fuzzy compensator," Fuzzy Systems, IEEE Transactions on, vol. 8, pp. 186-199, 2002.

[51] H. Medhaffar, et al., "A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator," International Journal of Modelling, Identification and Control, vol. 1, pp. 23-29, 2006.

[52] Y. Guo and P. Y. Woo, "An adaptive fuzzy sliding mode controller for robotic manipulators," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 33, pp. 149-159, 2003.

[53] C. M. Lin and C. F. Hsu, "Adaptive fuzzy sliding-mode control for induction servomotor systems," Energy Conversion, IEEE Transactions on, vol. 19, pp. 362-368, 2004.

[54] Iordanov, H. N., B. W. Surgenor, 1997. Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control, IEEE Trans. On control system technology, 5(2):254-260.

[55] Harashima F., Hashimoto H., and Maruyama K, 1986. Practical robust control of robot arm using variable structure system, IEEE conference, P.P:532-539.